高速計算プログラミングⅡ　第三回

M223337 田川幸汰

Q.

姫野ベンチのC言語OpenMP並列版（C + OMP, dynamic allocate version）をダウンロードし，gprof，Linux perf，VTune のいずれかを姫野ベンチに適用して結果を示せ．実行時のグリッドサイズは L とすること．

A.

今回は，gprofを姫野ベンチに適用した．関数ごとの情報であるflar profile部分の結果について図1に示す．

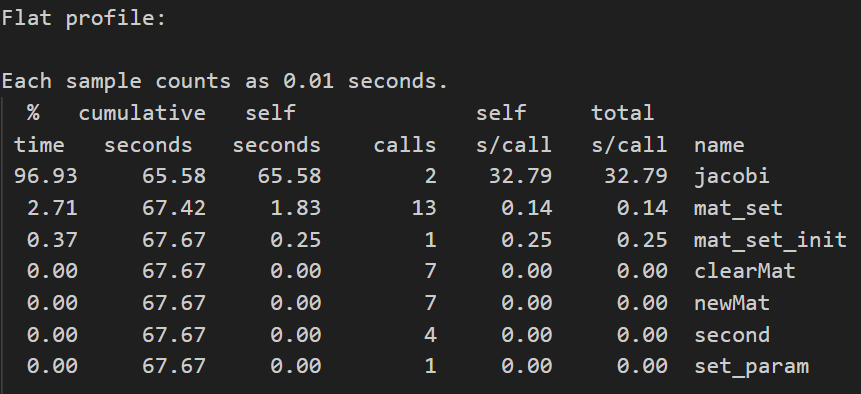


図１ gprof: flar profile部分

姫野ベンチの主な処理であるヤコビ反復法の実行時間は65.58秒、1回call当たりの実行時間は32.79秒となっている。またmat\_set関数は13回呼び出しされていて実行時間は1.83秒となっている。初期化などその他の関数の呼び出し時間はとても小さくなっている・

各関数とその呼び出し先関数 (callee)、呼び出し元関数(caller)の 情報であるcall graph部分の結果について図2に示す。また、図で表して可視化した様子を図3に示す。

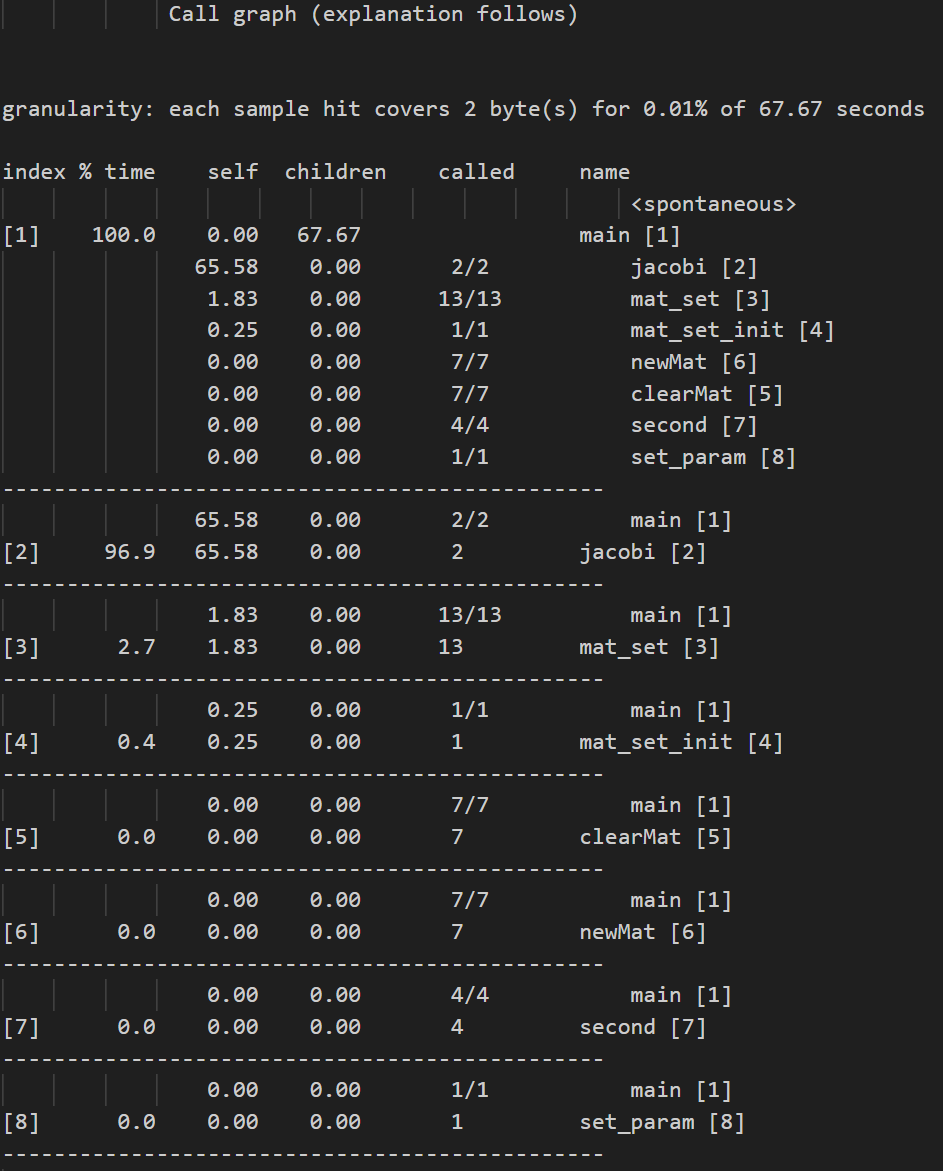


図2 gprof: call graph部分

タイムライン

自動的に生成された説明

図3 call graphの可視化

main関数で、行列のセット、ヤコビ反復法の処理が呼び出されていることがわかる。そして実行時間の約97％でヤコビ反復法の処理が行われていることがわかる。